Conference on Language Resources and Evaluation (LREC-2004). Lissabon, Portugal. - European Language Resources Association (ELRA), 2004. pp. 259-262.

Concept-based Queries: Combining and Reusing Linguistic Corpus Formats and Query Languages

Felix Sasaki*, Andreas Witt*, Dafydd Gibbon†, Thorsten Trippel†

*Universität Bielefeld, Fakultät für Linguistik und Literaturwissenschaft
– Computerlinguistik und Texttechnologie –
{felix.sasaki, andreas.witt}@uni-bielefeld.de
†Universität Bielefeld, Fakultät für Linguistik und Literaturwissenschaft
– Computational Linguistics and Spoken Language –
{ttrippel, gibbon}@spectrum.uni-bielefeld.de

Abstract

This paper proposes a methodology for querying linguistic data represented in different corpus formats. Examples of the need for queries over such heterogeneous resources are the corpus-based analysis of multimodal phenomena like the interaction of gestures and prosodic features, or syntax-related phenomena like information structure which exceed the expressive power of a tree-centered corpus format. Query languages (QLs) currently under development are strongly connected to corpus formats, like the *NITE Object Model* (NOM, Carletta et al., 2003) or the *Meta-Annotation Infrastructure for ATLAS* (MAIA, Laprun and Fiscus, 2002). The parallel development of linguistic query languages and corpus formats is due to the fact that general purpose query languages like XQuery (Boag et al., 2003) do not fulfill the changing needs of linguistically motivated queries, e.g. to give access to (non-)hierarchically organized, theory and language dependent annotations of multi modal signals and/or text. This leads to the problem that existing corpus formats and query languages are hard to reuse. They have to be re-developed and re-implemented time-consumingly and expensively for unforeseen tasks. This paper describes an approach for overcoming these problems and a sample application.

1. Background

1.1. Annotation formats and query languages

XML is the standard document description language which is nowadays supported in some way by almost all language resource projects. XML constrains the modelling process so that only tree-structured representations can be expressed directly: each element except the root element must have exactly one parent element. Consequently, items bracketed by elements cannot overlap, which leads to problems, for instance in dialogue description. In XML tag sets are defined and text is anntoted according to the tag set. The tag set can (and should) be defined formally in a schema language, e.g. DTDs, XML SCHEMA, or RELAX NG.

1.1.1. XML and XQuery

For querying XML, the World Wide Web Consortium is developing a standard language, analogous to SQL for relational databases, *XQuery* (Boag et al., 2003), using the potential of the *XPath data model* (Fernandez et al., 2003) to navigate the trees described by XML document instances. Applications supporting this standard may include the information gained from the schema, i.e. type information and legal nesting. XQuery as such does not presuppose the existence of a schema, but only the well-formedness of the instance. Consequently, it is possible to create queries which will never match anything, because the sequence of nodes specified in the query expression cannot be licensed by the schema. Currently available implementations of XQuery do not support the information gained by the schema, some[1] validate the document instance against a schema before the query is performed. Restrictions to the query which deviate from paths, i.e. to tree fragments in a given context that have certain properties such as a certain content or substructure, are limited to specific information given in the instance. Type information which generalises over classes of substructures or content cannot be used. Finally, XQuery is inadequate for querying non-tree structured information, because it relies on the data model of XML, which enforces proper nesting of annotation units, leading to a single hierarchy.

1.1.2. NITE and NXTSearch

The European NITE project, and its predecessor MATE, developed an XML-based representation for language data. A primary annotation level is used, in documentations of MATE and NITE usually the word level. Other layers which can exist are linked with this primary layer. An example is the annotation of the prepositional phrase in the sentence "The cat sat on the mat.". The first and primary annotation is the `w`ord annotation:

```
<s><w id='w1'>The</w> <w id='w2'>cat</w>
<w id='w3'>sat</w> <w id='w4'>on</w>
<w id='w5'>the</w> <w id='w6'>mat</w>.</s>
```

The abstract representation of this data results in a tree structure: Each element `w` has one parent element `s`.

The annotation of the prepositional phrase is created by introducing a new layer. In this layer an empty XML element for annotating chunks is used. This element points to a range of tokens in the sentence, e.g.:[2]

```
<ch id='chunk1' type='pp'
    href='#id(w4)..id(w5)'/>
```

The abstract integrated representation of the annotations results in a structure which is not representable as a tree:

[1] For example the Tamino XML database by Software AG.

[2] This example uses the syntax of MATE. NITE developed a different syntax, which is more focussed on the representation of time-aligned data.

The word elements with the ids `4`, `5`, and `6` have two parents, `s` and the element `ch`. Within the data model of NITE this representation is possible because it supports multiple intersecting hierarchies. Consequently, the query language of NITE (NiteQL) which is implemented in the NXT-Search Toolkit allows for quering of more complex structures than those permitted by XQuery.

1.1.3. AGs

The *Annotation Graph Model* (Bird and Liberman, 2001) defines a formal model for the interpretation of annotations in graph form. In the domain of speech annoation, this is done by the use of a reference to the timeline as a constant. For multiple annotations of primary data (Witt, 2002) a similar annotation model is achieved by using a primary level of annotation as an anchoring level. For example, the sequence of characters in textual data can supply the information about absolute ordering which is necessary for the anchoring.

Part of the annotation graph model is a formal description of a query language for these graphs, cf. Bird et al., 2000, with a query syntax. This is a very general model describing the properties of a query for a given annotation graph. The query is described in terms of a restricted selection from a set of arcs of an annotation graph, taking into account relations between the different arcs, such as overlaps, sequence, etc. No implementation is given.

For a database programmer familiar with SQL the individual constructs seem obvious, but this is not true for practical research queries. Researchers do not necessarily have the graph structure of annotations in mind when formulating queries based on more than one level of annotation. Consequently an intermediate level of querying is needed where general queries are mapped onto the graph based queries that can be executed in various data structures.

1.2. Modelling

1.2.1. Document grammars

Document grammars are used for defining XML-elements and their attributes and content models. By defining a content model it is possible to constrain which elements must, may, or must not occur as the content of an element. Furthermore a content model constrains the sequence of the legitimate elements, the type of the attributes (and, depending on the schema language, the elements). In addition some schema languages allow for the definition of 'fixed' or default values of attributes.

An XML document which is validated against a document grammar is known to have a certain structure. This information can help to speed up the process of querying, for example because it is not necessary to search for a certain element in a certain substructure if the schema does not permit the element in this substructure. This option is apparently not used in implementations of QLs of this type.

1.2.2. Multiple annotations

There are several techniques available for annotating language and speech data on multiple layers. The approaches mentioned above (i.e. MATE/Nite and AGs) use a hyperlink-based technique, where layers are interconnected through links. An alternative to this is to use of multiple (e.g. stand-off) annotations of the same textual base (Witt, 2002). This approach offers a lot of advantages: It allows for structuring text according to multiple concurrent document grammars without workarounds. Furthermore additional annotations can be subsequently included without changing already established annotations. For example, the hyperlink-based technique could make it necessary to introduce new anchor elements in the primary annotation layer if a new layer should be introduced. In using multiple annotations, the layers are independent of each other. Nonetheless they are interrelated, namely via the text. This allows on the one hand for inferences of relations between different annotation layers (Bayerl et al., 2003). On the other hand relations between *levels* (see below) and layers can be expressed in a formal way.

2. The LCD approach

We claim that an LCD approach allows for querying and expressing relations both in a single layer and between multiple annotated layers. As a solution to the problems of corpus formats and query languages described above, we consequently propose an abstract, conceptual level called "Linguistic Concept Descriptions" (LCD) which is built on top of the existing formats and query languages. Linguistic concepts, i.e. linguistic categories, principles etc. are declared in an LCD, which is separated from the query mechanisms and which can be used to retrieve instances of a concept in corpora. The linguistic concepts are organized in disjoint sets: for a specific language, linguistic theory or domain, a separate LCD is created, for example an LCD which encompasses concepts for an HPSG-grammar fragment of Japanese. Since an LCD encompasses a closed set of concepts, a generic mapping function from an LCD to query expressions in various query languages and their underlying corpus formats can be created. There are several advantages of this approach:

1. If the mapping function has been specified once, there is no need for the end-user to deal with the underlying corpus formats or query languages.
2. Various corpus formats can be used simultaneously. As a result of a uniform query, instances are retrieved from data in different corpus formats.
3. Since the same set of concepts can be instantiated in several corpus formats, the need to develop new query languages can be reduced. For unforeseen use cases, i.e. the integration of new linguistic categories in the HPSG-grammar model,[3] existing corpus-data and query-tools can be reused.

3. Properties of an LCD

The properties of an LCD are described with reference to Figure 1. An LCD consists of a set of models, e.g. `PhrasalStructure` and concepts, e.g.

[3]It could be desirable to introduce a MORPH feature on lexical signs to treat various kinds of morphological phenomena that are irrelevant in syntax. See Krieger, 1993; Lüngen, 2002

Sentence. The concepts are arguments of predicates. That is, an LCD is a set of logical statements which can be written in a straightforward notation of triples. Concepts are arguments of the predicate partOf, e.g. Sentence partOf PhrasalStructure. In this way, disjointness of models becomes explicit. Subordinated concepts are arguments of the predicate subClassOf, e.g. Non-embeddedVerbalPhrase subClassOf VerbalPhrase. To be able to operationalize an LCD, only proper parts and subclasses are permitted so that the statements must not lead to a cyclic structure: e.g. a statement like Sentence subClassOf Sentence. is not allowed.

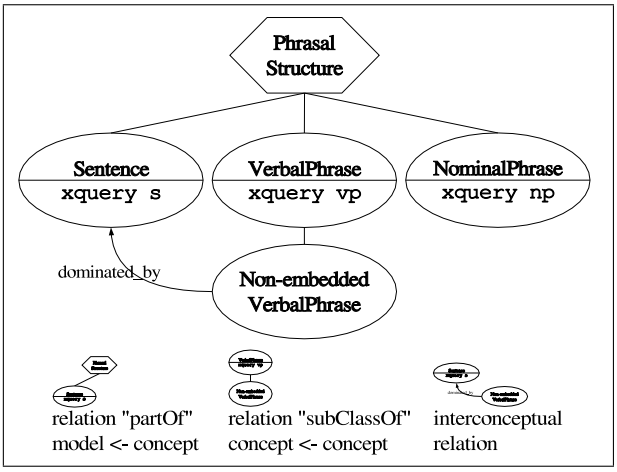

Figure 1: Properties of an Linguistic Concept Description.

In terms of Sowa, 1996 (cited after Fischer, 1998) this structure can be described as a *terminological ontology*. The statements create subtype, supertype and partOf relations. An important aspect of such an ontology is the *intensional and extensional notion of subsumption*. Intensionally speaking, there has to be a singular interpretation of the concept hierarchy; for example, a concept must not be subordinated to itself. Extensionally speaking, subordinated concepts share the properties, i.e. predicates of superordinated concepts, and the instances of the concepts. In the case of an LCD, the instances are to be found in corpus data. To create the extensional interpretation of an LCD, predicates are specified to define a mapping to query expressions in order to be able to access the corpus data.

3.1. Mapping to query expressions

The predicates describe a mapping to query expressions in various query languages. There are two kinds of predicates, top level predicates and other predicates. Top level predicates have three arguments. The first is a concept which is directly superordinated to a model, e.g. Sentence, VerbalPhrase or NominalPhrase. The second argument specifies the query language, e.g. XQuery. The third argument contains the query expression, e.g. for an s element.

The other predicates have the same two arguments to specify the concept and the query language. In addition, they have a finite set of other arguments to describe *interconceptual relations*. In Figure 1, such a relation is created via the predicate dominated_by. The statement is Non-embeddedVerbalPhrase dominated_by Sentence. As for the mapping to query expression, it depends on the query language and the corpus format in question how these predicates have to be interpreted. In the case of XQuery and an XML corpus with hierarchical annotations, dominated_by is interpreted as a step on the parent axis. For the concept Non-embeddedVerbalPhrase and XQuery, the query expression is vp[parent::s]. As for the Nite Query language *NiteQl*, the same predicate is interpreted as the dominance operator ^. The query expression is ($Sentence s) ($VerbalPhrase vp): $Sentence ^ $VerbalPhrase.

The application of a singular predicate must not lead to a query expression which combines several query langaguages. Nonetheless it is possible to combine different QLs. The combination of QLs is realised via the integration of query results in the instance documents, via a predefined XML namespace lcd. While executing the queries, attributes from this namespace are attached to nodes which are instances of a concept, for example lcd:VerbalPhrase=''someID''. To be able to use the result of a query from other query language, tests which match these attributes have to be specified. For example, instances of the concept Non-embeddedVerbalPhrase might have been retrived via NiteQl. An XPath expression could use these results with a test like //*[@lcd:VerbalPhrase].

3.2. Predefined predicates query expression mapping

To ease the task of creating an LCD, currently several predicates for the mapping to query expressions are defined. These encompass well-known basic structural relations in trees, complex paths in trees and multilayer-relations. The main relevant predicates for syntactic analysis are dominates and precedes, with counterparts like dominated_by. For complex paths in the (syntactic) tree-structure, so-called *caterpillar-expressions*, as defined by Brüggemann-Klein and Wood, 2000 are used in a separate set of predicates. The predicates take as one argument a name of a concept, and as the other a caterpillar-expression. The language of caterpillar-expressions consists of a finite set of moves and tests in the tree structure: up, first, last, left, right, isFirst, isLast, isLeaf, isRoot, the Kleene-star operator *, and a node-name test. The simplicity of these expressions makes the description of a generic mapping to queries straightforward. The caterpillar language has been implemented for example by Sasaki and Pönninghaus, 2003.

To be able to query non-hierarchical relations, multilayer-predicates as described in Bayerl et al., 2003 are defined. They take two concepts as arguments. To retrieve all sentences which exist just of one verbal phrase, the predicate identity can be applied in the statement i.e. Sentence identity VerbalPhrase. Such predicates are also used not for queries, but as a heuristic in the process of creating hierarchical-structured annotations from separate, textual annotations, see Witt et al., 2004. The predicates describe constraints which have to be ful-

filled in the result of the process. This application of the predicates demonstrates that the declarative description of linguistic concepts can be used for different purposes, specifically for querying and validation of constraints.

3.3. Representation of an LCD

The representation of an LCD makes use of the Resource Description Framework (RDF), and its extension RDF Schema (RDFS, Hayes and McBride, 2003). A set of linguistic concepts is represented with the constructs offerd by RDFS, like `rdfs:subClassOf`. Some of these constructs need an additional interpretation. For example, `rdfs:subClassOf` is used to specify `subClassOf` relations between concepts and `partOf` relations between concepts and models. RDFS is an extremely widely used resource format, permitting integration of an LCD with different resources.

4. Sample application and outlook

An LCD containing about 200 basic linguistic categories for Japanese has been developed.[4] In addition, an LCD containing categories of the HPSG-grammar fragment from the VERMOBIL-project (Kawata and Bartels, 2000) and another LCD of categories from a descriptive Japanese grammar which is used in a large-scale corpus (Kurohashi and Nagao., 2003) have been developed. Sample-corpus data is converted for flexibility into a Prolog-based corpus format which is queried with a dedicated query language. This format and the query language are described elsewhere (Sasaki et al., 2003); they are designed to allow for multi-level, non-hierarchical queries. In addition, other corpus data are represented in the NOM format and analyzed with the respective query language. The top level predicates are operationalized in an implementation of the MetaLex approach, cf. Trippel et al., 2004, for modelling co-referential phenomena in Japanese, cf. Sasaki and Witt, 2004.

A further perspective using abstractions over document grammars as a constraint mechanism for queries is currently being considered.

5. References

Bayerl, Petra Saskia, Harald Lüngen, Daniela Goecke, Andreas Witt, and Daniel Naber, 2003. Methods for the semantic analysis of document markup. In *Proceedings of the 2003 ACM symposium on Document engineering*. ACM Press.

Bird, S., P. Buneman, and W.C. Tan, 2000. Towards A Query Language for Annotation Graphs. In *Proceedings of LREC 2000*. Athens.

Bird, S. and M. Liberman, 2001. A formal framework for linguistic annotation. *Speech Communication*, (33 (1,2)):23–60.

Boag, S., D. Chamberlin, M. F. Fernández, D. Florescu, , and J. Siméon, 2003. XQuery 1.0: An XML Query Language. Technical report, World Wide Web Consortium.

Brüggemann-Klein, A. and D. Wood, 2000. Caterpillars: A Context Specification Technique. *Markup Languages: Theory and Practice*, 2(1):81–106.

Carletta, J., J. Kilgour, T. O'Donnell, S. Evert, and H. Voormann, 2003. The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML)*.

Fernandez, M., A. Malhotra, J. Marsh, M. Nagy, and N. Walsh, 2003. XQuery 1.0 and XPath 2.0 Data Model. URL: http://www.w3.org/TR/xpath-datamodel/, visited February 2004. W3C Working Draft 12 November 2003.

Fischer, D., 1998. From Thesauri towards Ontologies? In W. M. el Hadi, J. Maniez, and St. A. Pollit (eds.), *Structures and Relations in Knowledge Organization. Proceedings of the 5th ISKO-Conference, Lille*. Würzburg: Ergon Verlag.

Hayes, P. and B. McBride, 2003. RDF Semantics. Technical report, World Wide Web Consortium.

Kawata, Y. and J. Bartels, 2000. Stylebook for the Japanese Treebank in VERBMOBIL. Technical report, Verbmobil.

Krieger, H.-U., 1993. Derivation without lexical rules. Research Report RR-93-27, DFKI, Saarbrücken.

Kurohashi, S. and M. Nagao., 2003. Building a Japanese Parsed Corpus while improving the Parsing System. In A. Abeillé (ed.), *Building and Using Parsed Corpora*. Kluwer, Dordrecht.

Laprun, C. and J. Fiscus, 2002. Recent Improvements to the ATLAS Architecture. In *Proceedings of HLT 2002*. San Diego, California.

Lüngen, H., 2002. *A hierarchical model of German morphology in a spoken language lexicon environment*. Ph.D. thesis, Universität Bielefeld.

Sasaki, F. and J. Pönninghaus, 2003. Testing Structural Properties in Textual Data: Beyond Document Grammars. *Literary and Linguistic Computing*, 18(1):89–100.

Sasaki, F., A. Witt, and D. Metzing, 2003. Declarations of Relations, Differences and Transformations between Theory-specific Treebanks: A New Methodology. In J. Nivre (ed.), *The Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*. Växjö University, Sweden.

Sasaki, Felix and Andreas Witt, 2004. Co-reference in japanese task-oriented dialogues : A contribution to the development of language-specific and general annotation schemes and resources. In *Proceedings of LREC 2004*. Lisbon: ELRA.

Sowa, J. F., 1996. Ontologies for Knowledge Sharing. Manuscript of the invited talk at TKE 96.

Trippel, Thorsten, Felix Sasaki, and Dafydd Gibbon, 2004. Consistent storage of metadata in inference lexica: the metalex approach. In *Proceedings of LREC 2004*. Lisbon: ELRA.

Witt, A., 2002. Meaning and interpretation of concurrent markup. In *Proceedings of ALLC / ACH 2002*. Tübingen, Germany.

Witt, A., H. Lüngen, F. Sasaki, and D. Goecke, 2004. Unification of XML Documents with Concurrent Markup. In *Proceedings of ALLC / ACH 2004*. Gothenburg, Sweden.

[4] http://coli.lili.uni-bielefeld.de/Texttechnologie/Forschergruppe/sekimo/internet-praesentation/klassifikation/index.html